REPLACEMENT PAGES FOR SUBSTITUTE SPECIFICATION OF U.S. PATENT APPLICATION 09/301,438 ATTORNEY DOCKET NO. NS-3799 US

SUMMARY

[0006] The Digital Versatile Disc (DVD) and Digital Video Broadcast (DVB) decoder, in accordance with one embodiment of the present invention, includes a stream demultiplexer, a data storage buffer and a control Central Processing Unit (CPU). The stream demultiplexer demultiplexes and depacketizes the raw DVD data stream, subsequently stores the video and audio data in the storage buffer and informs the CPU thereabout.

[0007] The stream demultiplexer extracts the timing information embedded in each data pack of a data stream and, accordingly, generates messages about the stored data and their locations in the storage buffer. These messages include tags containing the decode time stamp, the presentation time stamp and the storage location of each data pack stored in the buffer.

[0008] The CPU reads the information recorded in each tag and, accordingly, generates task definition packets for use by the decoder.

[0009] The decoder is fully synchronized with respect to a signal generated by a video output processor of the system. Accordingly, in a steady state and under normal operating conditions, the CPU is interrupted only when the synchronization signal arrives thereby significantly reducing the number of interrupts that the CPU must service. Furthermore, the CPU benefits from having an entire period of the synchronization signal to service the interrupts and therefore has relaxed timing requirements.

[0010] The decoder is pipelined to reduce the idle time and hence to increase the utilization of the CPU and the video decoder. Accordingly, during each cycle of the synchronization signal, while the video decoder decodes the data, the CPU generates a task definition packet for the data to be decoded during the next cycle.

[0011] Because the stream demultiplexer removes all the timing information from the data before storing it in the storage buffer, the tasks performed by the audio and video decoders are simplified. Furthermore, because the CPU is the component that makes all the decisions about the particular data and its decode time, the audio and video decoders have vastly simplified tasks.

[0012] The stream demultiplexer also depacketizes the audio data stream. The stream demultiplexer extracts the timing information from each audio data pack and stores its payload in the buffer. Thereafter, the audio decoder determines the offset between the beginning of an audio data packet and the beginning of an audio frame and supplies the information to the CPU.

system header 202 have arrived. Timer 30 also includes facilities which trigger events when a particular time is reached. Consequently, timer 30 generates events based on specific times or records the times when specific events occur.

[0038] As is seen from Fig. 1, Video Output Processor (VOP) 40 receives a signal VsyncPhase from timer 30. VOP 40 generates a synchronization signal, called Vsync, every 16 milliseconds (i.e., 60 Hz for NTSC systems) which is the typical rate at which a Cathode Ray Tube (CRT) or a TV monitor is refreshed. Signal Vsync is supplied to Central Processing Unit (CPU) 54, video decoder 36 and timer 30. Signal Vsync synchronizes the activities of the various components of decoder 20 and, accordingly, controls the DVD data consumption rate, as is described below.

[0039] Whenever SD 26 extracts the pack header 201 of a data pack 200, it instructs timer 30 to store the current time in register 56. SD 26 stores the clock reference from the pack header in register 56. Therefore, register 56 stores both the clock reference of the data pack 200 as well as the time at which the pack arrives at SD 26. CPU 54 has access to register 56 and can read its content to determine the difference between the two timing data. In a steady state, when the system is fully synchronized, a fixed timing difference exists between the clock reference of data pack 200 and the time when data pack 200 arrives at SD 26.

[0040] Furthermore, after demultiplexing and depacketizing a DVD data pack 200, SD 26 extracts the associated time stamps (i.e., DTS 210 and PTS 212) of the data packet 206, stores video payload data 214 in buffer 48 and generates a video message, or tag, which contains the DTS and the address of the stored data in buffer 48. In other words, for each data pack 200, SD 26 generates a tag containing information about both the decode time stamp as well as the address of the stored data in buffer 48. The generation of the tags is a significant advantage of decoder 20.

[0041] The video messages, or tags, thus generated are made available to CPU 54 for further synchronization and control of decoder 20. Using the information stored in a tag, CPU 54 generates a Task Definition Packet (TDP) containing instructions to video decoder 36 about the location of the data in buffer 48. Upon the arrival of the next Vsync signal, video decoder 36 decodes the data identified by the TDP. If no TDP has been generated by CPU 54, video decoder 36 does not decode any data at the occurrence of the Vsync signal.

[0050] In MPEG, an audio time stamp is located at the beginning of a PES packet, but it actually refers to the first byte of the first audio sync frame that begins in the packet, and thus, there is no alignment between audio sync frames and PES packet boundaries. The audio bit streams contain start code patterns (sync words), but they are not individually uniquely identifiable from the actual audio data, thereby making it difficult to detect the start of an audio data frame.

[0051] In MPEG, a new audio frame is identified by identifying an audio marker that constitutes a repeating 12-bit field. Thus, for example, if the 12-bit pattern of an audio marker is identified three times, then it may be safely assumed, with a high degree of certainty, that the frame is an audio data frame.

[0052] After depacketizing an audio frame, SD 26 stores the PES payload in buffer 48 and generates an audio message, or tag, informing CPU 54 of the location of the stored audio frame in buffer 48. Consequently, SD 26 has knowledge of both the location of the audio frame in buffer 48 as well as its corresponding time stamp. SD 26, however, does not know the offset between the beginning of the packet and that of the frame.

[0053] Audio decoder 34 decodes the data and upon detecting the corresponding sync word, marks the address and so informs the CPU 54. Figure 4 shows the format of the message generated by audio decoder 34 to CPU 54 when audio decoder 34 finds a sync word in the compressed audio stream. Each such message has 32 bits. Bits [31:20] of each word are reserved, and only the lower 20 bits are used. The various parameters of the message when decoding MPEG audio are defined below:

NCHANS	number of audio channels in the input bit-stream (including LFE)
SAMPRATE	sampling rate
LAYER	1=MPEG layer 1, 2=MPEG layer 2
INBUF_LEVEL	Level (count) of the compressed audio data input buffer
BITRATE	Bit rate in kilobits/sec
FRM_SIZE	Size of the last decoded frame
OUTBUF_WR_PTR	Current write pointer of the output buffer used to store decoded audio data

channel DAC and/or an S/PDIF output for AC-3 playback, and an 8052 microcontroller for front panel, infrared (IR) and general purpose input/output (I/O).

[0074] Fig. 6 shows SD 26 connections to other modules in decoder 20 of Fig. 1 in accordance with one embodiment of the present invention. In particular, SD 26 connects to timer 30, a descrambler 28, a memory management unit (MMU) 60, a host bus interface 62, and a network port/DVD controller 64.

[0075] Referring to Fig. 6, as part of depacketizing and transporting the byte stream, SD 26 writes the extracted information to MMU 60. MMU 60 subsequently transfers the information into the system memory (e.g., buffer 48 of Fig. 1). Also, the audio decoder 34 of Fig. 1 and the video decoder 36 of Fig. 1 request audio and video data, respectively, from the system memory (e.g., buffer 48 of Fig. 1) using MMU 60. In one embodiment, the system memory includes thirty-two separate buffers (e.g., buffer 48 includes thirty-two separate sub-buffers): one buffer each for MPEG video, audio, sub-picture, teletext, and various other data and control streams. Further, SD 26 tags certain events in the bit stream being written to system memory with, for example, the desired decode or presentation time, the current write location in system memory, etc. SD 26 presents the tags to CPU 54 through the message queue (e.g., an event FIFO) stored in buffer 48 of Fig. 1.

[0076] For example, when SD 26 detects clock reference (CR) fields in the input stream, SD 26 provides the value to timer 30. Timer 30 may use this data to set the current system time when decoder 20 of Fig. 1 is attempting to gain initial synchronization. When the system (decoder 20) is already synchronized, SD 26 can capture the current time from the timer 30 when SD 26 detects CR fields and store the CR fields in registers for presentation to CPU 54.

[0077] SD 26 extracts system time stamps, either PCR for transport streams or SCR for program streams, and puts these extracted time stamps in host-accessible registers along with the current system time and the level of each buffer in the received packet. Using the time stamp and buffer level versus system time data as phase measurements into a software-controlled phase-locked loop, CPU 54 can adjust the frequency of the video pixel clock and the audio oversampling clock so that the system time base matches the source material time base on average. The adjustment will lower the rate of skip/repeat events in the presentation processes. When splices occur in a transport stream, there may be a discontinuity between time stamps on either side of the splice.